FalseMath: Evaluating Mathematical Reasoning of LLMs using Expectation Failure

Anonymous ACL submission

Abstract

Mathematical word problem solving is a popular method for evaluating the ability of Large Language Models (LLMs) to handle mathematical reasoning. LLMs have demonstrated notable proficiency in this domain. This paper introduces an innovative approach to evaluating the reasoning capabilities of LLMs, employing the paradigm of expectation failure to unearth reasoning gaps. Utilizing a rubric that emphasizes conceptual clarity over mathematical prowess, and a newly curated dataset named FalseMath, comprising 500 intentionally flawed word problems (partially obtained through LLM augmentation), we demonstrate through experiments that LLMs have yet to attain complete conceptual mastery in the art of algebraic word problem reasoning.

1 Introduction

Recent advancements in language models (LLMs) have shown remarkable progress in various natural language processing tasks, including mathematical word problem solving. As Natural Language Processing (NLP) seeks to equip machines with the capability to comprehend and respond to natural language, the ability to solve word problems reflects a model's proficiency in contextual understanding, logical reasoning, and semantic comprehension. Word problems often require a deep understanding of language nuances, inference, and real-world context—challenges central to NLP. This paper delves into the landscape of LLMs, with a particular focus on their capabilities in handling mathematical word problems. In the realm of LLMs, algebraic word problem solving has been hailed as a "solved" problem, as evidenced by the accuracy rate exceeding 90% demonstrated in the GPT-4 report (OpenAI, 2023) on GSM8k (Cobbe et al., 2021), a high-quality popular dataset. However, this paper challenges the prevailing notion, asserting that there is a subtle yet pervasive undercurrent of flaws in mathematical reasoning within this framework. We advocate for a more discerning evaluation approach by introducing the mechanism of expectation failure.

Numerous datasets have been suggested and continue to emerge for the word problem-solving task. Examples include GSM8k (Cobbe et al., 2021), GHOSTS (Frieder et al., 2023), SVAMP (Patel et al., 2021), CONIC10K (Wu et al., 2023), CHAMP (Mao et al., 2023), MATH (Hendrycks et al., 2021), among others. These datasets progressively escalate in difficulty. We argue that, despite the apparent success in solving algebraic word problems, there remain unexplored gaps in mathematical reasoning that warrant further investigation.

Question	A two digit number is twice its reverse. The sum of the digits is 20. Find the number.
GPT-3.5	198
GPT-4	82

Figure 1: Analysing GPT on Math Word Problems. On posing an impossible word problem, GPT3.5 and GPT 4 provide incorrect answer

Illustrated through a case study (Figure 1) highlighting a deliberate error introduced into a simple algebraic word problem, we showcase how expectation failure brings forth nuanced insights into the limitations of LLMs. By presenting models with queries resembling their training data but intentionally incorporating errors, we unveil their tendency to neglect subtleties. This underscores the necessity for a deeper comprehension of mathematical contexts.

Furthermore, we explore the potential of using LLM hallucination to create false datasets for evaluation purposes. By modeling expectation failure through hallucination, we ease the development of challenge datasets. We also describe a comprehensive five-point rubric for evaluation, aligning with

Domain	Concept	Example with Math Error
Number	Place values and limits	Give me an example of a four digit number whose sum of digits is 40
Percentage	Fractions and values	If p% of x is more than x, prove that p < 100
Age	Years, time, relation	Carmen is 12 years older than David. Five years ago, the sum of their ages was 28. When will Carmen's age be half of David's age?
Mixtures	Concentration, percentage, limits	A mixture containing 6% boric acid is to be mixed with 2 quarts of a mixture which is 15% acid in order to obtain a solution which is 4% acid. How much of the 6% solution must be used?
Flow problems	Speed, against/with flow, current	A boat travels 30 km up a river in the same time it takes to travel 50 km down the same river. Find the speed of the current of the river if it takes more time downstream than upstream to cover the same distance.

Table 1: Design Principles of FalseMath

the insights garnered through expectation failure. The rubric, designed to assess concept understanding and mathematical accuracy, provides a holistic framework for evaluating LLMs in the intricate domain of algebraic word problem solving. Additionally, we propose the integration of self-verification prompting as a means to excel in these metrics.

Our contributions include:

- The introduction of a challenge dataset, FalseMath
- A robust evaluation rubric for conceptual clarity
- Experimental evaluation of GPT on FalseMath

In summary, we challenge the prevailing notion of algebraic word problem-solving as a "solved" problem by LLMs. Through the lens of expectation failure, we expose the flaws in mathematical reasoning, and not focus on solving alone.

2 Expectation Failure for Evaluation

The concept of expectation failure as a challenge has been often used in NLP. A case in point is the employment of linguistic challenges like garden path sentences (Jurayj et al., 2022) for parsing. Garden path sentences are structurally ambiguous phrases that mislead initial parsing attempts, often leading to incorrect interpretations. Consider the classic example, "The old man the boats," where the sentence initially directs readers to interpret "old man" as the subject, only to require a reinterpretation when the intended subject becomes clear.

In a similar vein, we employ a method of developing challenging word problems that are mathematically nonsensical. What we endeavor to accomplish is to dig out, errors in reasoning that may not have been captured while designing a system to aggressively solve a word problem. Similar challenge datasets such as SVAMP (Patel et al., 2021) employ the same principle but SVAMP still maintains the framework of having question both linguistically and mathematically sound. In our case, we present a set of word problems that are linguistically correct but *intentionally* mathematically incorrect. As such, it does not serve as a resource for training purposes.

To create the challenge algebraic word problem set FalseMath, we selected five commonly found sub-topics: numbers, percentages, age-related problems, mixtures, and upstream/downstream problems. A domain expert (person with math pedagogical experience) was enlisted to generate erroneous word problems that reveal a concept misconception. Examples are described in Table 1.

The resultant core set comprises 60 word problems, each deliberately imbued with errors. This core set is further enriched through augmentation

using ChatGPT, a process that will be expounded upon in the subsequent section.

2.1 Hallucination for Augmentation

Hallucination (Ji et al., 2023) in the well-studied phenomenon of large language models generating sentences which include falsities, recurrent rambling on the same point and so on. While this fact hampers automated natural language augmentation as it slows down the process and often involves expensive annotators for fact-checking. In our application, hallucination actually becomes our friend, as we want to introduce as many mathematical inconsistencies as possible. After a domain expert puts the initial effort in designing such erroneous word problems, GPT3.5 was used to expand this core dataset to 500. The table statistics are described in Table 2.

Dataset	Source	Size
FalseMathCore	Human	60
FalseMath	Human + ChatGPT	500
	Domain	**Size**
	Number	133
	Age	147
	Percent	83
	Mixture	24
	Flow	79
	Misc	34

Table 2: FalseMath Statistics

3 Non-Benchmark Evaluation

When developing an evaluation resource for word-problem solving, the standard modus-operandi is to provide a set of word problems and the corresponding aspirational solution, be it a piece of text, the final numerical answer, the associated equations and so on. By using a dataset of erroneous word problems, there is no one-size-fits-all approach in dataset design that may be used to model the ideal answer. Hence, we instead developed a five-point rubric for qualitative evaluation.

3.1 Evaluation Rubric

The qualitative method of evaluating generated text is commonly used for complex text evaluation (Frieder et al., 2023). Given in Table 3 is the evaluation mechanism employed. The points that can be given ranges from 1 to 5, with 1 being the lowest and 5 being the highest. The rubric is structured in such a way that lower scores signify flawed conceptual understanding. This design allows for a focus on emphasizing conceptual clarity over mere mathematical accuracy. Also, the score 4 is unique in the sense that we would like to reward cautious solutions over aggressive wrong solutions.

Point	Explanation
1	Concept Wrong, Math Wrong
2	Concept Wrong, Math Right
3	Concept Right, Math Wrong
4	Concept Right, Math Unattempted
5	Concept Right, Math Right

Table 3: FalseMath Rubric

3.2 Self-Verification Prompting

Self-verification is a popular prompting strategy (Zhou et al., 2023) that suggests the system check the work cautiously. We found the tendency to aggressively solve the word problem, most likely a product of the training data and test set, often leads the system to make mathematical inaccuracies. For this purpose, we used the following prompt as shown in Figure 2 in a section of our experiments.

> Let us solve word problems in a concise, formal way with the following principles.
>
> Enumerate the concepts involved
> Explain the work
> Display the answer as "The answer is"
> Check your work
>
> With these principles, show the 4 steps for the following question :

Figure 2: Self-Verification Prompt

4 Experimental Analysis

For the experiments, we utilize the popular LLMs GPT 3.5 and GPT-4 (OpenAI, 2023). The implementation was done with Python3, OpenAI API and executed on Google Colab. The evaluation involves two datasets: FalseMathCore and FalseMath. Using the rubric, automated evaluation by GPT-4 was performed. Additionally, a domain expert (male, Asian) with experience in teaching pedagogy evaluated FalseMathCore. Subsequently, we examine the impact of self-verification prompting on both datasets.

5 Discussion

The investigation reveals that algebraic word problem solving is NOT a solved problem. GPT-4 does

Input	Human	GPT-4
FalseMathCore		
GPT-3.5	3.23	3.13
GPT-3.5 w Prompt	3.22	3.23
GPT-4	**3.96**	**4.56**
GPT-4 w Prompt	3.84	4.33
FalseMath		
GPT-3.5	-	3.48
GPT-3.5 w Prompt	-	3.57
GPT-4	-	**4.53**
GPT-4 w Prompt	-	4.46

Table 4: FalseMath Evaluation - Two datasets FalseMathCore (60) and FalseMath (500) are evaluated on GPT3.5 and 4 using the rubric designed by the actors - human and GPT4 - (higher the score, the better)

perform significantly better than GPT 3.5. It is interesting to note that in the GPT4 evaluation, not having a prompt improved performance slightly. The prompting strategies seem to have a small effect on the performance. This suggests that self-verification alone did not suffice to regulate the reasoning process. However, neither is GPT3.5 nor GPT4, a perfect math reasoning model for simple word problems (as would be evidenced by a perfect score of 5), suggesting some caution to be exercised while deploying these models into homework solution bots or tutoring systems. While human and GPT-4 evaluations are consistent for GPT3.5 models, there is a marked difference when it comes to GPT-4. This observation also implies that the method of GPT-4 evaluation may not be reliable. Also, we have deliberately not provided a sample answer because we do not want to bias algorithms to perform better on this dataset. We believe the gold rush to beat leaderboards for math word problem solving has also led to focus on solving, than reasoning. On close examination, we found that the same word problem would be solved correctly when the number is 4 digit long, but not when the number is 8 digit long. The (in)stability of the answers is also a well-studied problem (Frieder et al., 2023). Rather, we want to use these design principles of expectation failure to incrementally test the math reasoning capabilities of such models.

6 Related Work

Automatic math word problem solving has been an active area of research in the past decade ((Lu et al., 2023), (Liu, 2023)). Some of the most recent works (Gao et al., 2023),(Kim et al., 2022),(Wang and Lu, 2023), (Zhao et al., 2023), (Schick et al., 2023), (Xie et al., 2023), (Zheng et al., 2023) use a judicious mix of GPT-4 and prompting and code generation to navigate the difficult realm of mathematical word problem solving. There has also been significant work on the strengths and limitations of LLMs on reasoning ((Huang and Chang, 2023), (Tan et al., 2023), (Gaur and Saunshi, 2023) etc). The community is vibrantly and actively looking into both math word problem solving and math reasoning of LLMs (Ferreira, 2023) and hence, it is possible we might have missed individual citations on the same. The focus of this contribution is to provide a more nuanced perspective of the math reasoning capabilities of LLMs and not to take any level of mathematical prowess for granted.

7 Conclusion

This study has explored the complexities of algebraic word problem-solving, questioning the common belief that it is a fully resolved challenge for LLMs. By utilizing expectation failure, we propose that incorporating mathematical errors in the format of the training data reveals errors in methodological reasoning. Beyond traditional datasets, we have introduced *FalseMath*—a dataset intentionally featuring mathematically erroneous word problems. The experiment was designed with a core set of 60 word problems and augmented to 500, by taking advantage of ChatGPT hallucination for augmentation. We then introduced a five point rubric for evaluation, that placed a higher premium on concept understanding, rather than word problem solving. The evaluation of GPT 3.5 and 4 reveal that all in not well in the world of simple math word problem solving. While GPT-4 is certainly significantly better than GPT-3.5, we examined the errors to uncover inconsistencies in the face of many red herrings that resemble the training data. This research can be extended to assess various categories of LLMs, conduct a more detailed examination of instances where the models falter, enlarge the primary dataset by engaging additional experts, complete human evaluation of the entire dataset, and replicate the methodology in other facets of mathematical reasoning. By adopting the design principle of expectation failure and employing the evaluation rubric of FalseMath, the study aims to push the frontiers of evaluating mathematical NLP reasoning tasks, fostering a more nuanced and comprehensive understanding of language models' proficiency in mathematical reasoning.

Limitations

This work presents FalseMath - a method to evaluate LLM math reasoning systems through the method of testing expectation failures. The dataset presented is small and the design of it is based on two annotators. This work can be made more robust by adding more annotators and building a bigger core. The evaluation metric often necessitates human evaluation, though we have demonstrated GPT4 evaluation. The confidence of GPT4 evaluation is yet to be examined.

References

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Patrícia Ferreira. 2023. Automatic dialog flow extraction and guidance. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 112–122, Dubrovnik, Croatia. Association for Computational Linguistics.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Christian Petersen, Alexis Chevalier, and Julius Berner. 2023. Mathematical capabilities of chatgpt. *arXiv preprint arXiv:2301.13867*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Vedant Gaur and Nikunj Saunshi. 2023. Reasoning in large language models through symbolic math word problems. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5889–5903, Toronto, Canada. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

William Jurayj, William Rudman, and Carsten Eickhoff. 2022. Garden path traversal in GPT-2. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 305–313, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Bugeun Kim, Kyung Seo Ki, Sangkyu Rhim, and Gahgene Gweon. 2022. Ept-x: An expression-pointer transformer model that generates explanations for numbers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4442–4458.

Lei Liu. 2023. Processing advantages of end-weight. In *Proceedings of the Society for Computation in Linguistics 2023*, pages 250–258, Amherst, MA. Association for Computational Linguistics.

Pan Lu, Liang Qiu, Wenhao Yu, Sean Welleck, and Kai-Wei Chang. 2023. A survey of deep learning for mathematical reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14605–14631, Toronto, Canada. Association for Computational Linguistics.

Yujun Mao, Yoon Kim, and Yilun Zhou. 2023. WAMP: A competition-level dataset for assessing the mathematical reasoning capabilities of LLMs. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*.

OpenAI. 2023. Gpt-4 technical report.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada. Association for Computational Linguistics.

Tianduo Wang and Wei Lu. 2023. Learning multi-step reasoning by solving arithmetic tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1229–1238.

Haoyi Wu, Wenyang Hui, Yezeng Chen, Weiqi Wu, Kewei Tu, and Yi Zhou. 2023. Conic10K: A challenging math problem understanding and reasoning dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6444–6458, Singapore. Association for Computational Linguistics.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. 2023. Decomposition enhances reasoning via self-evaluation guided decoding. *arXiv preprint arXiv:2305.00633*.

Xu Zhao, Yuxi Xie, Kenji Kawaguchi, Junxian He, and Qizhe Xie. 2023. Automatic model selection with large language models for reasoning. *arXiv preprint arXiv:2305.14333*.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023. Progressive-hint prompting improves reasoning in large language models. *arXiv preprint arXiv:2304.09797*.

Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, et al. 2023. Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification. *arXiv preprint arXiv:2308.07921*.